

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-6396

(43) 公開日 平成9年(1997)1月10日

(51) Int.Cl. ⁸	識別記号	庁内整理番号	F I	技術表示箇所
G 1 0 L	9/14		G 1 0 L 9/14	G
	9/18			J
H 0 3 M	7/30	9382-5K	H 0 3 M 7/30	E
				B

審査請求 未請求 請求項の数17 O L (全 13 頁)

(21) 出願番号 特願平7-150550

(22) 出願日 平成7年(1995)6月16日

(71) 出願人 000004226

日本電信電話株式会社

東京都新宿区西新宿三丁目19番2号

(72) 発明者 大室 伸

東京都千代田区内幸町1丁目1番6号 日

本電信電話株式会社内

(72) 発明者 池戸 丈太郎

東京都千代田区内幸町1丁目1番6号 日

本電信電話株式会社内

(72) 発明者 守谷 健弘

東京都千代田区内幸町1丁目1番6号 日

本電信電話株式会社内

(74) 代理人 弁理士 草野 卓

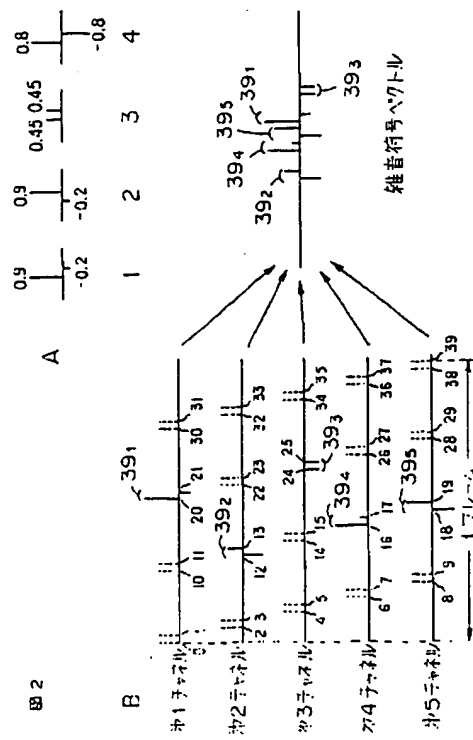
最終頁に続く

(54) 【発明の名称】 音響信号符号化方法及び音響信号復号化方法

(57) 【要約】

【目的】 CELP方式音声符号化で雑音符号ベクトル探索を効率的に行うことを可能とする。

【構成】 符号帳32₁～32₅にそれぞれ連続する2サンプルのパターンベクトルの4種類(図2A)を記憶し、その各1つのパターンベクトル39₁～39₅をそれぞれ選択し、ベクトル39₁～39₅は1フレーム(40サンプル点)内の取得位置は各4個所とし、互いに異なるサンプル点とする(図2B中の点線)。ベクトル39₁～39₅の和ベクトルを雑音符号ベクトル41とし、再生音声の歪が最小となるベクトル41を探索する。



【特許請求の範囲】

【請求項1】 適応符号帳から、フレームにあるいはサブフレーム（以下両者を総称してフレーム単位と記す）単位に取り出した過去の駆動音源ベクトルを、ピッチ周期に対応する周期により繰り返して作成した時系列ベクトル、雑音符号帳から取り出した雑音符号ベクトルの時系列ベクトル、あるいはこれらの両方より得られる時系列ベクトルにより、合成フィルタを駆動して音響信号を再生して入力音響信号ベクトルに対する歪が最小になるように上記適応符号帳、上記雑音符号帳の選択を行う音響の符号化方法において、

少なくとも2サンプルからなるパターンベクトルを少なくとも1つ格納したパターン符号帳から1つのパターンベクトルを取出し、現在のフレームの先頭を基準位置として、そのフレーム内で基準位置から時間方向に、事前に決められた複数ヶ所のサンプル位置のうちのひとつの位置までシフトして立てることによって得られるベクトルを上記雑音符号ベクトルとする、

ことを特徴とする音響信号符号化方法。

【請求項2】 請求項1に記載の音響信号符号化方法において、

上記パターン符号帳から取出した上記パターンベクトルに、正負の符号をつけて上記雑音符号ベクトルを表現する、

ことを特徴とする音響信号符号化方法。

【請求項3】 請求項1または請求項2に記載の音響信号符号化方法において、

上記ひとつのパターンベクトルを、ある位置までシフトして得られるベクトルを1チャンネルとして、

複数のチャンネルから得られるベクトルの和ベクトルを、上記雑音符号ベクトルとする、

ことを特徴とする音響信号符号化方法。

【請求項4】 請求項3に記載の音響信号符号化方法において、

あるチャンネルにおける、パターンベクトルのシフトするサンプル位置を、他のチャンネルにおいてパターンベクトルのシフトするサンプル位置のいずれとも重複しないように設定する、

ことを特徴とする音響信号符号化方法。

【請求項5】 請求項3または請求項4に記載の音響信号符号化方法において、

上記複数のチャンネルの中から、2あるいは複数のチャンネルを選択し、

その選択したチャンネルのうち、1つあるいは複数のチャンネルのパターン情報およびシフト位置情報を出力しないことにより、情報を送信するチャンネルを切り換える、

ことを特徴とする音響信号符号化方法。

【請求項6】 請求項5に記載の音響信号符号化方法において、

上記選択したチャンネルのパターン情報が同じである

ことを特徴とする音響信号符号化方法。

【請求項7】 請求項3乃至6の何れかに記載の音響信号符号化方法において、

フレーム内の入力音響信号と対応した信号のサンプル列の時間配列を逆とした時系列ベクトルを、上記合成フィルタに通した、あるいはその合成フィルタのインパルス応答と畳み込んだ時系列ベクトルを用いて、各チャンネルにおけるシフト位置の候補をより少ない数の候補に予備選択する、

ことを特徴とする音響信号符号化方法。

【請求項8】 請求項3乃至6の何れかに記載の音響信号符号化方法において、

各チャンネルにおいてとりうる、パターンベクトルの候補と、シフト位置の候補のすべての組み合わせから、1チャンネル分の駆動ベクトル候補を生成し、

フレーム内の入力音響信号と対応する信号のサンプル信号の時間的配列を逆方向にした時系列ベクトルを、上記合成フィルタに通した、あるいは上記合成フィルタのインパルス応答と畳み込んだ時系列ベクトルと上記生成した駆動ベクトル候補との内積の絶対値、または内積の二乗値、あるいは、上記内積の絶対値を上記生成駆動ベクトル候補のパワーで除した値によって、候補の組み合わせを予備選択する、

ことを特徴とする音響信号符号化方法。

【請求項9】 請求項7に記載の音響信号符号化方法において、

各チャンネルにおいてとりうる、パターンベクトルの候補と、上記予備選択後のシフト位置の候補のすべての組み合わせから、1チャンネル分の駆動ベクトル候補を生成し、

フレーム内の入力音響信号と対応した信号のサンプル信号の時間的配列を逆とした時系列ベクトルを、上記合成フィルタに通した、あるいはその合成フィルタのインパルス応答と畳み込んだ時系列ベクトルと上記駆動ベクトル候補との内積の絶対値、または内積の二乗値、あるいは、上記内積の絶対値を上記駆動ベクトル候補のパワーで除した値によって、候補の組み合わせを予備選択する、

ことを特徴とする音響信号符号化方法。

【請求項10】 請求項8または請求項9に記載の音響信号符号化方法において、

パターンベクトルとシフト位置の組み合わせを予備選択したときの尺度の値のうち、各チャンネルにおける第一位の候補の尺度の値を、チャンネル相互で比較して、値の大きい順に2あるいは数チャンネル選択し、

その選択されたチャンネルから生成される駆動音源信号を、上記合成フィルタを通して合成した波形ベクトルの、入力音響信号ベクトルに対する歪みが最小になるように、当該チャンネルにおけるパターンベクトルとシフト位置を決定し、

残りのチャンネルの全部、または一部から生成される駆動音源信号を、先に決定したチャンネルによって生成される駆動音源信号に加算し、上記合成フィルタを通して合成した波形ベクトルの、入力音響信号ベクトルに対する歪みが最小になるように、当該チャンネルにおけるパターンベクトルとシフト位置を決定する、
ことを特徴とする音響信号符号化方法。

【請求項11】 請求項1から請求項10までのいずれかに記載の音響信号符号化方法において、
生成された駆動音源信号を、上記合成フィルタを通して合成した波形ベクトルの入力音響信号ベクトルに対する歪みを計算する過程で、
上記合成フィルタのインパルス応答を要素とする行列と、その転置行列との積を先に計算してメモリに展開しておき、このメモリの記憶を用いて計算する際に、
上記積行列の要素をその数行または数列おきに値を保持して、保持しない行または列の要素の値は、保持している値で近似的に置き換えて計算する、
ことを特徴とする音響信号符号化方法。

【請求項12】 請求項1乃至11の何れかの音響信号符号化方法において、
上記パターンベクトルは2サンプルよりなる、
ことを特徴とする音響信号符号化方法。

【請求項13】 適応符号帳から、フレーム又はサブフレーム（以下フレームと総称する）単位で過去の駆動音源ベクトルを、入力周期符号に応じた周期で繰返した時系列ベクトル、雑音符号ベクトルの時系列ベクトルあるいはこれら両時系列ベクトルにより合成フィルタを駆動して音響信号を再生する音響信号復号化方法において、
それぞれ少なくとも2サンプルからなるパターンベクトルをそれぞれ少なくとも1つ格納した複数のパターン符号帳を設け、
これらパターン符号帳のそれぞれについて入力パターン符号及び位置符号の組に応じてそのパターン符号帳からパターンベクトルを取り出し、フレーム上の位置に配置して上記雑音符号ベクトルを作成する、
ことを特徴とする音響信号復号化方法。

【請求項14】 請求項13に記載の音響信号復号化方法において、
入力されるパターン符号及び位置符号の組数より多くのパターン符号帳を設け、入力符号帳選択符号に応じて、上記パターン符号帳を選択して復号に用いる、
ことを特徴とする音響信号復号化方法。

【請求項15】 請求項13に記載の音響信号復号化方法において、
上記パターン符号帳ごとに対応位置符号に応じてパターンベクトルを配置すべきフレーム上の位置を異ならせておき、少なくとも1つのパターン符号帳については予め決められたパターンベクトル配置位置の組を複数有し、これら複数のパターンベクトル配置位置の組の1つを入力

選択符号に応じて用いる、
ことを特徴とする音響信号復号化方法。

【請求項16】 請求項13乃至15の何れかに記載の音響信号復号化方法において、
上記パターン符号帳から取出された各パターンベクトルをその上記組に含まれている入力サイン符号に応じて逆性を制御して上記配置を行う、
ことを特徴とする音響信号復号化方法。

【請求項17】 請求項13乃至16の何れかに記載の音響信号復号化方法において、
上記パターンベクトルは2サンプルよりなる、
ことを特徴とする音響信号復号化方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】この発明は、音声または音楽などの音響信号のスペクトル包絡特性を表すフィルタを、音源ベクトルで駆動して音響信号を合成することを利用した予測符号化により、音響信号系列を少ない情報量でデジタル符号化する高能率音響信号符号化方法及びその復号化方法に関する。

【0002】

【従来の技術】デジタル移動体通信において、電波を効率的に利用したり、音声信号または音楽信号の蓄積サービス等で記憶媒体を効率的に利用するために、高能率音響信号符号化方法が用いられる。現在、音声信号を高能率に符号化する方法として、原音声信号をフレームまたはサブフレーム（以下フレームと総称する）と呼ばれる5～50ms程度の一定間隔の区間に分割し、その1フレームの音声信号を、周波数スペクトルの包絡特性を表す線形フィルタの特性と、その線形フィルタを駆動するための駆動音源信号との2つの情報に分離し、それぞれを符号化する手法が提案されている。この手法において、駆動音源信号を符号化する方法として、音声信号のピッチ周期（基本周波数）に対応すると考えられる周期成分と、それ以外の成分とに分離して符号化する方法が知られている。この駆動音源情報の符号化法の例として、符号駆動線形予測符号化（Code-Excited Linear Prediction: CELP）がある。この技術の詳細については、文献 M.R. Schroeder and B.S. Atal, "Code-Excited Linear Prediction (CELP): High Quality Speech at Very Low Bit Rates", IEEE Proc. ICASSP-85, pp. 937-940, 1985に記載されている。

【0003】図6に上記従来の符号化方法の構成例を示す。入力端子1に入力された原音声信号（入力音声信号）は、線形予測分析部2において、入力音声信号の間波数スペクトル包絡特性を表す線形予測パラメータが計算される。得られた線形予測パラメータは線形予測パラメータ符号化部3において、符号化されて線形予測パラメータ復号化部4に送られる。線形予測パラメータ復号化部4では、受け取った符号からフィルタ係数を再生

し、合成フィルタ5および歪み計算部6に送る。なお、線形予測分析の詳細および線形予測パラメータの符号化例については、例えば古井貞熙著“デジタル音声処理”(東海大学出版会)に記載されている。ここで、線形予測分析部2、線形予測パラメータ符号化部3、線形予測パラメータ復号化部4および合成フィルタ5は非線形なものに置き換えてもよい。

【0004】適応符号帳7からはそのバッファに記憶された直前の過去の駆動音源ベクトル(既に量子化された直前の1〜数フレーム分の駆動音源ベクトル)を入力された周期符号に相当する長さで切り出し、その切り出したベクトルをフレームの長さになるまで繰り返すことによって、入力音声信号の周期成分に対応する時系列ベクトルの候補が出力される。

【0005】雑音符号帳8からは、音声信号の非周期成分に対応する1フレーム分の長さの時系列符号ベクトル(雑音符号ベクトル)の候補が出力される。これらの候補は入力音声信号とは独立に符号化のためのビット数に応じてあらかじめ指定された数の候補雑音符号ベクトルが記憶されている。適応符号帳7および雑音符号帳8から出力された各時系列ベクトルの候補は、乗算部9、10においてそれぞれ重み符号帳11において作成された重みが乗算され、加算部12において加算されて駆動音源ベクトルの候補となる。

【0006】合成フィルタ5は、線形予測パラメータ復号化部4の出力をフィルタの係数とする線形フィルタで、加算部12の出力である駆動音源ベクトル候補を

$$d = ||x_p - y_p||^2$$

を歪み評価尺度計算部19で計算し、歪み評価尺度dが最小となるような合成信号ベクトルyを選択することによって、最適な周期符号、雑音符号、重み符号が選択される。聴覚重みづけフィルタ16, 17を用いないこともある。このとき、すべての周期符号、雑音符号、重み符号の組み合わせから、尺度dが最小になるものを選択するのが、再生音声信号の歪みを小さくするうえで望ましいが、実際には演算処理量の問題から、周期符号、雑音符号、重み符号の順に決めることも多い。重み符号を、適応符号と雑音符号にかけるそれぞれの重みを、一括してベクトル量子化しない場合には、周期符号、適応符号重み、雑音符号、雑音符号重みの順に決定してもよい。重みを一括してベクトル量子化する場合でも、適応符号重みを、暫定的に非量子化値(最適値)に設定しておいて、周期符号、非量子化適応符号重み、雑音符号、適応符号重みと雑音符号重みのベクトル量子化、の順に決めても良い。また、各符号帳における選択を決める際に、ただひとつの符号に決定してしまうのではなく、いくつもの候補を残しておいて、それらの最適な組み合わせに決定してもよい。

力として再生音声信号の候補を出力する。合成フィルタ5の次数すなわち線形予測分析の次数は、一般に10〜16次程度が用いられることが多い。なお、既に述べたように、合成フィルタ5は非線形なフィルタでもよい。

【0007】歪み計算部6では、合成フィルタ5の出力である再生音声信号の候補と、入力音声信号との歪みを計算する。この歪みの計算は、例えば聴覚重み付けなど、合成フィルタ5の係数または量子化していない線形予測係数を考慮にいれて行なうことが多い。符号帳検索制御部13では、各再生音声信号候補の入力音声信号に対する歪みが最小となるような周期符号、雑音符号および重み符号を選択し、そのフレームにおける駆動音源ベクトルを決定する。

【0008】符号帳検索制御部13において決定された周期符号、雑音符号、重み符号と、線形予測パラメータ符号化部3の出力である線形予測パラメータ符号は、符号送出部14に送られ、利用の形態に応じて記憶装置に記憶されるか、または通信路を介して受信側へ送られるために出力される。図6中の歪み計算部6の構成例を図7に図6と対応する部分と同一符号を付けて示す。入力音声信号である入力時系列ベクトルxと、合成フィルタ5からの再生音声信号候補である合成信号ベクトルyの間の歪みをはかる尺度の一例として、入力時系列ベクトルxを聴覚重みづけフィルタ16に通したベクトルx_pから、合成信号ベクトルyを聴覚重みづけフィルタ17に通したベクトルy_pを引算部18で差し引いたベクトルx_p - y_pから、歪み評価尺度として、

(1)

【0009】図8Aは、図7の構成例を、等価な形で書き直したものである。合成フィルタ5、聴覚重みづけフィルタ17は線形のフィルタであるから、図7に示すように、適応符号ベクトルと雑音符号ベクトルを、別々に合成フィルタ21、聴覚重みづけフィルタ22と合成フィルタ23、聴覚重みづけフィルタ24に通して、それぞれを、入力音声時系列ベクトルxを聴覚重みづけフィルタ16を通したx_pから差し引いて歪み評価尺度を計算することができる。

【0010】図8Aにおける合成フィルタ23と、聴覚重みづけフィルタ24を合わせて、図8Bに示すように等価なFIR型フィルタ25で実現することができる。インパルス応答算出部26では、フィルタ23, 24を合わせた特性のフィルタ25のインパルス応答を計算し、このインパルス応答と雑音符号ベクトルCに重みg_rをかけたベクトルg_rcとの畳み込みをフィルタ25で行うことによって、等価な線形フィルタを実現できる。このとき、インパルス応答算出部26から出力されるインパルス応答を、h₀, h₁, h₂, ..., h_{N-1} (Nは1フレームのサンプル数)として、h_jを

$$H = \begin{bmatrix} h_0 & 0 & 0 & 0 & \cdots & 0 \\ h_1 & h_0 & 0 & 0 & \cdots & 0 \\ h_2 & h_1 & h_0 & 0 & \cdots & 0 \\ h_3 & h_2 & h_1 & h_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ h_{N-1} & h_{N-2} & h_{N-3} & h_{N-4} & \cdots & h_0 \end{bmatrix} \quad (2)$$

とおき、これをインパルス応答行列と呼ぶ。適応符号ベクトルと適応符号重みが、先にまたは暫定的に決まっているものとし、適応符号ベクトルに重みをかけ、図7Aにおいて、合成フィルタ21および聴覚重みづけフィル

$$d = \| r - g_r H c \|^2$$

に等しい。このとき、 g_r は雑音符号のあとで決定するため、ここでは任意の値をとりうると仮定し、 g_r で

(3) 式を偏微分して、これをゼロとする g_r を計算

$$d' = (r^t H c)^2 / (c^t H^t H c) \quad (4)$$

(3) 式を最小にするような雑音符号ベクトル c を選ぶことは、(4) 式を最大にする雑音符号ベクトル c を選ぶことに等しい。

【0012】

【発明が解決しようとする課題】 CELP方式において、問題となるのは、低いビットレートで高い品質の再生音声を得ようとする（圧縮率をあげようとする）と、雑音符号帳に多くの雑音符号ベクトルを記憶するために、非常に多くのメモリを必要とするだけでなく、雑音符号帳出力に重みをかけ、合成フィルタを通して歪みを計算する一連の処理を、雑音符号帳に蓄えられる雑音符号ベクトルの数だけ繰り返す処理に、非常に多くの演算処理が必要となるという問題があった。

【0013】 この問題に対して、Algebraic Code-Excited Linear Prediction (ACELP) という方式が提案されている。この方式は、雑音符号帳にベクトルパターンを蓄えるのではなく、高さが1の正負パルスを、フレーム内に数本、例えば、40サンプルのフレームに対して、4本を適当な位置に立てることによって、雑音符号ベクトルを出力するものである。この方式では雑音符号ベクトルを記憶するためのメモリが不要で、パルスを立てる位置のテーブルのみをメモリに記憶すればよいという特徴を持つ。また、演算処理量に関しても、従来のCELP方式に比べると、パルスが数本しかないので、かなり少ない処理量で高い品質を実現できるというメリットがある。なお、この方式は各パルスの位置と極性をランダムに取らせることにより雑音符号ベクトルとして作用させるものであるが、ACELP方式の詳細は、例えば、文献、R. Salami, C. Laflamme, and J-P. Adou, "8 kbit/s ACELP Coding of Speech with 10 ms Speech-Frame: a Candidate for CCITT Standardization", IEEE Proc. ICASSP-94, pp. II-97 に記載されている。

【0014】 しかしながら、ACELP方式をもってしても、実時間で符号化処理を実現するためには、高価な

タ22を通した信号ベクトルを、引算部28で x_p から差し引いた信号、つまり入力音声信号ベクトルから同期成分を除去した信号を r とおく。このとき、(1) 式で示される評価尺度 d は、

(3)

し、その g_r を (3) 式に代入すると (4) 式となる。

【0011】

デジタルシグナルプロセッサ (DSP) が必要であり、安価なプロセッサで実現するためには、さらに演算量の少ない方式でなければならない。ACELP方式は、非常に簡単なモデル化で高品質な再生音声信号を実現するという、すぐれた方式であるけれども、モデルが簡単なゆえに、上記文献で提案されているよりも処理量の低減を試みると、急速に再生音声信号の品質が劣化するという問題が生じる。

【0015】 この発明の目的は、安価なプロセッサで許容される範囲内の少ないメモリ量、かつ非常に少ない演算量で、高品質な再生音声信号を得ることを可能とする音響信号の符号化方法、及びその復号化方法を提供することにある。

【0016】

【課題を解決するための手段】 この発明の符号化方法では、雑音符号ベクトルとして、数サンプルを単位とするパターンベクトルを、フレーム内でシフトして、フレーム内に数個立てることによって雑音符号ベクトルを表現し、パターンベクトルの形状と、シフト位置を効率的に探索することによって、少ない演算処理で、高い品質の再生音を得る音声の符号化方法を実現する。また、パターンベクトルを記憶するためのメモリは、従来のCELP方式に比べて、非常に少なくてもよい。一方、ACELP方式に比べると、より実際の音声の特徴にあったパターンベクトルを用意し、効率的にシフト位置を探索することによって、演算量がより少なく、高品質な符号化方法を実現する。

【0017】 この発明の復号化方法では数サンプルを単位とするパターンベクトルをパターン符号帳に記憶しておき、入力されたパターン符号に応じてパターンベクトルを取出し、これを入力されたその位置符号に応じてフレーム内に配置して雑音符号ベクトルを再生する。

【0018】

【実施例】 図1にこの発明の符号化方法の実施例を適用

した符号化装置の機能的構成例を示し、図6と対応する部分に同一符号を付けてある。この発明では雑音符号帳8の代りに雑音符号帳31が設けられる。雑音符号帳31には複数のパターン符号帳32₁～32_Mが設けられ、これら各パターン符号帳32₁～32_Mにはそれぞれ数サンプルからなるパターンベクトルが1乃至10程度記憶されている。つまり各パターンベクトルは例えば連続するnサンプル点からなり、そのnサンプル点以外のサンプル点はゼロである。符号帳32₁～32_Mは保持するパターンベクトルは共通のものとしてもよい。これらパターン符号帳32₁～32_Mはそれぞれ符号帳検索制御部13からのパターン符号33₁～33_Mにより指定されてパターンベクトルが取出される。これら取出されたパターンベクトルはそれぞれ乗算部34₁～34_Mで符号帳検索制御部13からのサイン符号35₁～35_Mに応じて+1又は-1が乗算され、つまり極性が制御される。これら乗算部34₁～34_Mの出力はシフト部36₁～36_Mで符号帳検索制御部13からの位置符

$$t = nMj + (k-1)n; \quad j = 0, 1, 2, \dots, S \quad (5)$$

で示される位置に配置するようにする。Mはチャンネル数、nはパターンベクトルのサンプル数である。この場合はあるサンプル点は、いずれか1チャンネルでのみ表現され、複数のパターンベクトルが重なることはない。すなわち、各チャンネルのシフト部から出力されるベクトルは、チャンネル相互で0以外の値を同一サンプル点にもたない。ここで、Sは、フレーム(サブフレーム)長Nと、n、Mによって決まる値である。雑音符号としては、各チャンネルにおけるパターンベクトルのインデックス(パターン符号)と、シフト位置インデックス(位置符号)と、サイン符号とが出力される。

【0020】上記構成例によって、雑音符号帳31を構成するとき、場合によっては、あるチャンネルで表現できるパターンベクトルおよびシフト位置から生成されるベクトルのいずれを使用しても、そのチャンネルから出力されるベクトルを使用しない場合よりも、歪みが増える

(歪み評価尺度(4)式が小さくなる)ことがある。その場合には、当該チャンネルを使用しないことを表すコード(符号)を用意し、復号側でそのコードを受け取ったときには、当該チャンネルから出力されるベクトルを加算しないことにすると、再生品質が劣化しない。また、ビットレートの制限によっては、Mチャンネル分の情報を送信または蓄積できないことがある。その場合には、チャンネル数Mを減らすことによって、ビットレートを下げられるが、単純にMを小さくするのではなく、Mはそのままにして、第eチャンネルと第fチャンネルのいずれかを歪みの低減に効果的なチャンネルを符号化フレームごとに選択して切り替えることによって、実質的なチャンネル数を減らしてビットレートを下げる一方、再生音の品質劣化を抑えることができる。

【0021】以下に、n=2サンプル、M=5チャネ

号37₁～37_Mに応じてフレーム内の位置に配置される。シフト部36₁～36_Mの各出力は加算されて1フレームの雑音符号ベクトルとして乗算部10へ出力される。つまり、1フレーム分のバッファに対し各極性制御されたパターンベクトルが位置符号に応じたサンプル点に配置され、その際重なる時は加算され、そのバッファから雑音符号ベクトルが出力される。つまりパターン符号により決められるパターンベクトルを位置符号によりフレーム上に配置したパターンベクトルを生成するチャンネルをM個有し、そのMチャンネルよりのパターンベクトルの和により雑音符号ベクトルを生成する。各パターンベクトルのnサンプル点は、連続しないでとびとびのサンプル点でもよい。

【0019】シフト部36₁～36_Mでの配置位置は相互に重なってもよく、例えば3サンプルパターンベクトルの二つがその最後のサンプルと最初のサンプルとが重なるように配置してもよい。しかし例えば、第kチャンネルでは、

ル、N=40サンプル、S=4個所、ベクトルパターンは各チャンネル4パターンとした場合について具体的に説明する。各パターン符号帳32₁～32₅には、2サンプルからなる4種類のパターンベクトルが格納される。この4パターンの形状は、代数的に、例えば、{+1, 0}、{0, +1}、{+1, +1}、{+1, -1}のように決めてもよい。この決定は例えば図1中の合成フィルタ5の入力パルスを観察し、発生頻度の多い2サンプルの組の状態から行う。より音声の特徴を表すように、学習手続き、すなわち、大量の音声データを符号化してみて、その歪みの総和が最小となるようなパターンを代表パターンとして用いるほうが、高い品質を実現することができる。この場合のパターンベクトルの例を図2Aに示す。パターン符号帳にはこの各二つのサンプル点の値が組として格納される。

【0022】この例における第1チャンネルのパターンの配置位置は40点のうち次の4個所の何れかとなり、

(0, 1), (10, 11), (20, 21), (30, 31)

第2チャンネルのパターンの配置位置は、次の4個所の何れかとなり、

(2, 3), (12, 13), (22, 23), (32, 33)

第3チャンネルのパターンの配置位置は、次の4個所の何れかとなり、

(4, 5), (14, 15), (24, 25), (34, 35)

第4、第5チャンネルのパターンも以下同様にして決る。つまり図2Bに示すように、第1チャンネルのパターンベクトル39₁は(20, 21)に配置されているか、点線で示す他の3個所の何れかを取ることができる。同様

に第2乃至第5チャンネルの各パターンベクトル $3\ 9_2 \sim 3\ 9_5$ はそれぞれ図に示している位置以外に点線で示す各3個所の何れかを取ることができる。第1乃至第5チャンネルの各パターンベクトルをフレーム上に配置してベクトル的に加えて雑音符号ベクトル $4\ 1$ が得られる。図2Bに示した例では各パターン符号帳 $3\ 2_1 \sim 3\ 2_5$ に互いに異なる4種類のパターンベクトルを格納した場合である。

【0024】この例で出力する情報は、チャンネルあたり、4種類のパターンの何れであるかを表すために2ビット、4個所の配置位置の何れであるかを表すために2ビット、パターンベクトルの正負を表すために1ビットの計5ビットとなる。5チャンネル分の全てを送出すると、1フレーム（サブフレーム）あたり、25ビット必要になる。ある特定のチャンネルから出力されたベクトルを使用しないためのコードを割り当てる場合には、これを示すビットを別途割り当ててもよいが、そのチャンネルにおいてあるパターンの、ある配置位置、ある正負符号の場合を例外として「使用しないためのコード」に割り当てると、ビットレートを上げずに、品質を向上させることができる。この場合、そのチャンネルにおいて本来そのコードに割り当てられていた形状を表現できなくなるが、それによる品質の劣化分よりも、品質の向上分のほうが大きいような形状コードを選ぶ。

【0025】実際には、フレーム長が40サンプルのときに、5チャンネルをフルに使用して25ビットを使用するのは、若干無駄である。その場合には、例えば、図3Aに示すように第3チャンネルのシフト部 $3\ 6_3$ の出力と第5チャンネルのシフト部 $3\ 6_5$ の出力とをスイッチ手段43で切替え、第1乃至第4チャンネルを使用する場合と、第1、第2、第4、第5チャンネルを使用する場合とを符号化フレームごとにより方を選択し、その何れを選択したかを示すビットを1つ加えることにより、4チャンネル分の出力符号の 4×5 ビットに1ビット加え、21ビットを符号化出力とすればよい。

【0026】更に図3Bに示すように、符号帳 $3\ 2_5$ を省略し、第3チャンネルのシフト部 $3\ 6_3$ と第5チャンネルのシフト部 $3\ 6_5$ とをスイッチ手段44で切替え、使用し、切替えのためのビットとして1ビット使用してフレームあたり21ビット出力としてもよい。また第3チャンネルと第4チャンネルの切替えを行うと共に更に例えば第2チャンネルと第4チャンネルを切り替えると、送信パターンとして第1、2、3チャンネルあるいは第1、3、4チャンネル、第1、2、5チャンネル、第1、4、5の何れかとなり、17ビットでよい。このような切り替えをする場合、隣接するチャンネルどうして切り替えてもよいが、2チャンネル以上離れたチャンネルと切り替えたほうが効果的である。また、例えば、第3チャンネルと第5チャンネルを切り替え、残った4つのチャンネルの中から、歪み低減に貢献度の高い3つのチャンネル分を選択して使用する方

法も効果的である。この切り替えの方法では、どのチャンネルを選択したかを表すのに、2ビット必要となり、合計（サブ）フレームあたり18ビット必要になるが、貢献度の低いチャンネルを捨てることで、17ビットの場合よりも高い品質を実現できる。

【0027】次に、少ない演算量で、効果的に最適な符号を探索する実施例を示す。演算量を問わなければ、5チャンネルの、すべてのベクトルパターン、配置位置、正負符号の組み合わせの中から、(4)式を最大にする組み合わせを選択すればよいが、それでは、ACELP方式よりも逆に演算量が増えてしまう。この発明のねらいは、ACELPよりも音声データを表現しやすいモデル化を行うことによって、演算量を削減しても、品質の劣化を生じない方法を実現するところにある。

【0028】以下に、図3Bに示した雑音符号帳 $3\ 1$ でフレームあたり21ビットに符号化の場合の一探索例を示す。

ステップ1. 第1予備選択として、第1、第2、第4チャンネルは4配置位置から、第3、第5チャンネルは切り替えのため合わせて8配置位置から、各3個所の位置を予備選択する。その選択のための尺度は、例えば $r^t\ H$ の各パターンベクトル配置位置（2サンプル）における絶対値の和の大きい順に選択するとよい。つまり $r^t\ H$ はフレームの各サンプル点と対応して、この例では40個の値をもっており、例えば第1チャンネルの配置位置0番目と1番目と対応して $r^t\ H$ の0番目と1番目の各値の絶対値の和をとる。以下同様にすべての配置位置について、前記絶対値の和を取り、これらすべての絶対値の和の大きい順に、第1、第2、第4チャンネルから各3つ、第3、第5チャンネルから3つの配置位置を選択する。この絶対値の和が大きいということは、フレーム中のそのサンプル点位置に、駆動音源ベクトルのパルスが立ちそうであることを示している。なお r^t は入力音声信号と対応した信号（図8A中の引算部28の出力）の時間軸を反転した時系列ベクトル、つまり信号 r のサンプル列の時間的配列を逆としたものである。

【0029】ステップ2. 第2予備選択として、第1予備選択で得られた各チャンネルについて3つの配置位置と、対応チャンネルの4パターンベクトルとの組み合わせた12個の候補の中から、4つのパターンと配置位置の組み合わせを予備選択する。第 k チャンネルのシフト部から出力されるベクトルを $c\ k$ とすると、この第2予備選択の選択尺度には、 $|r^t\ H\ c\ k| / c\ k^t\ c\ k$ を用いるとよい。この分子のみでもよいが、分母も含めたほうが大ききにより正規化され、予備選択による品質の低下を抑えることができる。なお、 $r^t\ H$ は第1予備選択で計算した値をそのまま利用すればよい。また、 $c\ k^t\ c\ k$ は、パターンベクトルのパワであるから、事前にメモリに展開しておけば、毎フレーム毎に計算する必要はない。 $H\ c\ k$ はパターンベクトル候補 $c\ k$ を合成フィルタ

25 (図8B)に通した時系列、また合成フィルタは25のインパルス応答と c_k とを畳み込んだ時系列ベクトルである。この時系列ベクトルと r^t との内積の絶対値の代りにその内積の二乗値を用いてもよい。

【0030】ステップ3. 第2予備選択で選ばれた各チャンネルの4つの候補のうち、各チャンネル毎の第1候補つまり尺度が最も大きかったものを取出し、その4つの第1候補を大きい順に並べ、並べ替えた順に対応チャンネルをチャンネル1'、2'、3'、4'とする。

ステップ4. チャンネル1'と2'の各4組の組み合わせ(16通り)の中から、(4)式を最大にする組み合わせを選択する。(4)式の分子の計算では、 $r^t H c_k$ は第2予備選択で用いた値をそのまま利用する。分母の計算では、各フレームに1回、 $H^t H$ のマトリクスを先に計算してメモリに展開し、 c_k のゼロでないサンプル点が関係する項のみを計算すればよい。なお、(サブ)フレーム長が40サンプルのとき、 $H^t H$ をそのままメモリに展開すると、 $40 \times 40 = 1600$ ワードのメモリが必要になるが、実際には、 $H^t H$ は対称行列であるため、左下または右上の三角形部分の値のみを記憶すれば、820ワードのメモリでよい。それでもメモリ(RAM)の容量が多すぎるときは、 $H^t H$ を左下三角行列にした後、例えば、21行目から24行目要素は、すべて25行目の値で代用するなど、 $H^t H$ の要素の一部をとびとびで保持するなどによって、メモリ量を低減することができる。実験の結果、21行目から35行目までを、5行おきに値を保持することによって、490ワードのメモリで、ほとんど品質が劣化しないことが確認された。また、この方法では、 $H^t H$ の計算に要する演算時間も少なくできる。これは $H^t H$ の各要素は隣り合っているものの値は比較的近い値であるが、離れている要素間では値が可成り異なるから、このような近似が可能となる。

【0031】ステップ5. チャンネル3'と4'の各4組の組み合わせ(16通り)の中から、(4)式を最大にする組み合わせを選択する。計算手順に関しては、ステップ4と同様である。(サブ)フレームあたり、18ビットまたは17ビットにする場合は、前出のようにチャンネルを切り替えて使用すれば、ビットレートを下げられるだけでなく、演算量も21ビットの場合にくらべて削減できる。なお、18ビットの場合のチャンネルの選択は、上記ステップ5で、チャンネル3'か4'のいずれかから、(4)式が大きくなるほうを選択して、他方を捨てることで選択する。

【0032】上述において、パターン符号帳は1つでもよく、つまり1チャンネルでもよく、つまり1、2、4、8、又は16程度とされる。また各パターン符号帳に格納するパターンベクトルは1つでもよい。図1では各パターン符号帳の出力パターンベクトルに対してサイン符号35₁～35_M、雑音符号帳はサイン符号35₁～35_Mを省略してもよい。各チャンネルの出力、つまり図1

中のシフト部36₁～36_Mの出力のベクトル和を求め、その際上述したようにゼロでないサンプル点が重なってもよい。この場合探索手順は前述と同様に行えばよいが、重なりを許すため探索位置が多くなり、それだけ処理量が多くなる。つまり、チャンネル間でパターンベクトルの配置位置が互いに重ならないようにすると、最適配置位置の探索を効率的に行うことができる。

【0033】上述の探索手順において、ステップ1の第1予備選択を省略してもよい。しかし第1予備選択をすればそれだけ最適配置位置の探索の効率が向上する。合成フィルタ25としてはFIR型フィルタに限られるものでない。次にこの発明による復号化方法の実施例を適用した復号化装置の機能構成図4を参照してこの発明の復号化方法を詳細に説明する。

【0034】入力端子51より符号化信号は入力分離部52で各符号に分離され、線形予測パラメータ符号はLPC係数レジスタ53に格納され、周期符号は周期レジスタ54に格納され、重み符号は重みレジスタ55に格納され、雑音符号、つまりパターン符号33₁～33_M、サイン符号35₁～35_M、位置符号37₁～37_Mが雑音符号レジスタ56に格納される。LPC係数レジスタ53内の線形予測パラメータ符号は線形予測パラメータ復号化部57で復号され、フィルタ係数が求められて合成フィルタ58に設定される。合成フィルタ58の駆動信号ベクトルは分岐されて適応符号帳59に入力される。適応符号帳59は周期レジスタ54内の周期符号に応じた長さの過去の駆動ベクトルが取出され、それが1フレーム分繰返されて乗算部61へ出力される。重みレジスタ55内の重み符号が復号化され、適応符号帳用重み、雑音符号帳用重みがそれぞれ乗算部61、62へ出力される。符号化装置のパターン符号帳32₁～32_Mと同一のパターン符号帳63₁～63_Mが設けられる。例えば各パターン符号帳63₁～63_Mに2サンプルのパターンベクトルが4つそれぞれ格納されている。これらパターン符号帳63₁～63_Mは対応チャンネルのパターン符号33₁～33_Mによりそれぞれパターンベクトルが取出される。これら取出されたパターンベクトルは乗算部64₁～64_Mでそれぞれサイン符号35₁～35_Mに応じて+1又は-1が乗算される。乗算部64₁～64_Mの出力パターンベクトルはそれぞれ配置部65₁～65_Mにより位置符号37₁～37_Mに応じたフレーム上の位置と対応して1フレーム分、例えば40ポイント分の駆動源バッファ66内に格納される。駆動源バッファ66は各フレームごとにシリアルに出力され、雑音符号ベクトルとして乗算部62へ供給され、重みが乗算された後、加算部67で乗算部61からの重み付けられた適応符号帳出力と加算され、駆動ベクトルとして合成フィルタ58へ供給される。

【0035】以上のようにして復号化信号が音声信号と

して復号化される。図3Aに示した雑音符号ベクトルの生成と対応する復号化方法を、その要部のみを示す図5Aを参照して説明する。雑音符号レジスタ56内にパターン符号 $3\ 3_3$ 、サイン符号 $3\ 5_3$ 、位置符号 $3\ 7_3$ 又はパターン符号 $3\ 3_5$ 、サイン符号 $3\ 5_5$ 、位置符号 $3\ 7_5$ が1チャンネル分として格納され、そのパターン符号によりパターン符号帳 $6\ 3_3$ 、 $6\ 3_5$ の両方からパターンベクトルが取出され、それらパターンベクトルが乗算部 $6\ 4_3$ 、 $6\ 4_5$ で前記1チャンネル分として格納されているサイン符号により $+1$ 又は -1 が乗算され、更に配置部 $6\ 5_3$ 、 $6\ 5_5$ により位置符号に応じてバッファ66に配置格納される。この際にレジスタ56中の符号帳切替えを示す1ビットの切替符号68により切替え手段69が制御され、符号化側で第3チャンネルが出力された場合は配置部 $6\ 5_3$ の出力側がバッファ66に接続され、第5チャンネルが出力された場合は配置部 $6\ 5_5$ の出力側がバッファ66に接続される。その他は図4の構成と同様である。

【0036】図3Bに示した雑音符号ベクトル生成と対応した復号化方法の要部を図5Bに示す。この場合は雑音符号レジスタ56に、第3チャンネルの符号として、パターン符号 $3\ 3_3$ とサイン符号 $3\ 5_3$ と、位置符号 $3\ 7_3$ 又は $3\ 7_5$ とその位置符号の何れかを示す選択符号71とが格納され、そのパターン符号 $3\ 3_3$ によりパターン符号帳 $6\ 3_3$ が読出され、その読出されたパターンベクトルは乗算部 $6\ 4_3$ でサイン符号 $3\ 5_3$ に応じて $+1$ 又は -1 が乗算され、その乗算されたベクトルは配置部 $6\ 5_3$ と $6\ 5_5$ とへ供給される。位置符号 $3\ 7_3$ 、 $3\ 7_5$ の何れが入力されても配置部 $6\ 5_3$ 、 $6\ 5_5$ の両者が制御され、配置部 $6\ 5_3$ 、 $6\ 5_5$ の出力側は選択符号71に応じて切替え手段72が制御され、符号化側でシフト部 $3\ 6_3$ が選択された場合は配置部 $6\ 5_3$ がバッファ66に接続され、シフト部 $3\ 6_5$ が選択された場合は配置部 $6\ 5_5$ がバッファ66に接続される。

【0037】復号化側におけるパターン符号帳に対する条件は符号化側と同一とすることで自づと定まる。上述では主として音声信号の符号化、復号化にこの発明を適用したが音楽信号の符号化、復号化にも適用できる。

【0038】

【発明の効果】この発明を利用した場合と、ACELP方式による場合とで、雑音符号ベクトル探索にかかる演算時間と再生音声の品質(SN比)を比較したものを図5Cに示す。演算時間は、ワークステーションで、シミュレーションプログラムを実行したときに要したCPU時間である(H^t Hの演算時間は含まない)。この結

果、ACELP方式に比べても、非常に少ない演算量で、高い品質を実現できることが確認された。

【0039】以上述べたように、この発明の符号化方法によれば複数のサンプル点よりなるパターンベクトルを、フレーム上のどの位置に配置するかにより雑音符号ベクトルを生成するため、このパターンベクトルをくずさず、それだけ制限され、従来のACELP方式のように各1個のパルスがフレーム上すべての位置をとる場合と比較して探索数が少なくても済む。

【0040】特に合成フィルタの駆動音源ベクトルは、連続する2対のパルスが多いことから、連続する2サンプルのパターンベクトルを用いる場合は整合性が高く、予備選択を導入しても、誤選択となるおそれが少なく、少ない演算量で正しい符号化、つまり良品の符号化を行うことができる。この発明の復号化方法によれば、この発明の符号化方法により効率的な探索を行った符号を復号化でき、品質の良い音響信号を再生することができる。

【図面の簡単な説明】

【図1】請求項3の発明による符号化方法を適用した符号化装置の機能構成例を示すブロック図。

【図2】Aはパターンベクトルの具体例を示す図、Bはこの発明方法における各チャンネルのパターンベクトルとその取り得るフレーム上の位置との例、その和ベクトルである雑音符号ベクトルを示す図である。

【図3】Aは請求項5の発明の符号化方法を適用した雑音符号帳31の例を示すブロック図、Bは請求項6の発明の符号化方法を適用した雑音符号帳31の例を示すブロック図である。

【図4】請求項13の発明の復号化方法を適用した復号化装置の機能構成例を示すブロック図。

【図5】Aは請求項14の発明の復号化方法を適用した復号化装置の雑音符号ベクトル再生部の要部を示すブロック図、Bは請求項15の発明の復号化方法を適用した復号化装置の雑音符号ベクトル再生部の要部を示すブロック図、Cはこの発明の符号化方法、従来のACELPによる雑音符号ベクトル探索処理時間を示す図である。

【図6】従来のCELP方式の符号化装置を示す機能構成図。

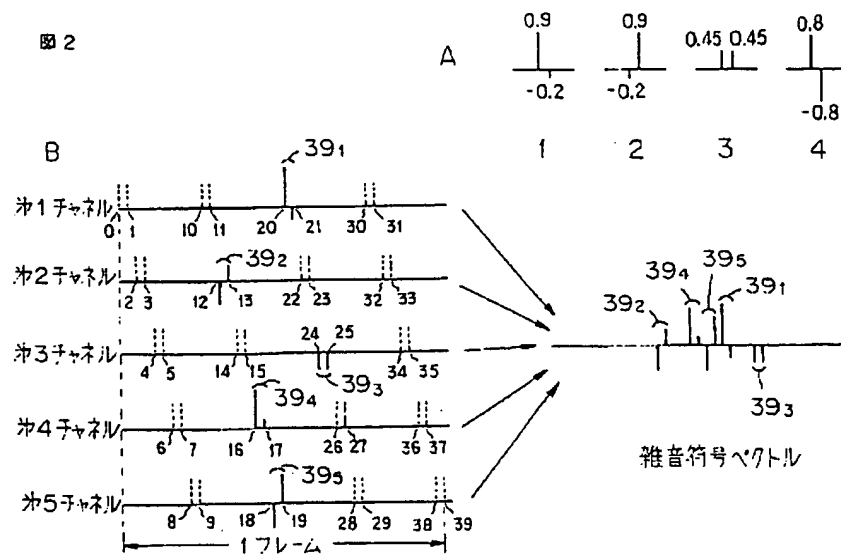
【図7】図6中の歪み計算部6の詳細を示すブロック図。

【図8】Aは図7中の駆動音源に関する部分を等価な形で置き換えたブロック図、Bは図8A中のフィルタ23、24を1つのフィルタで実現した例を示すブロック図である。

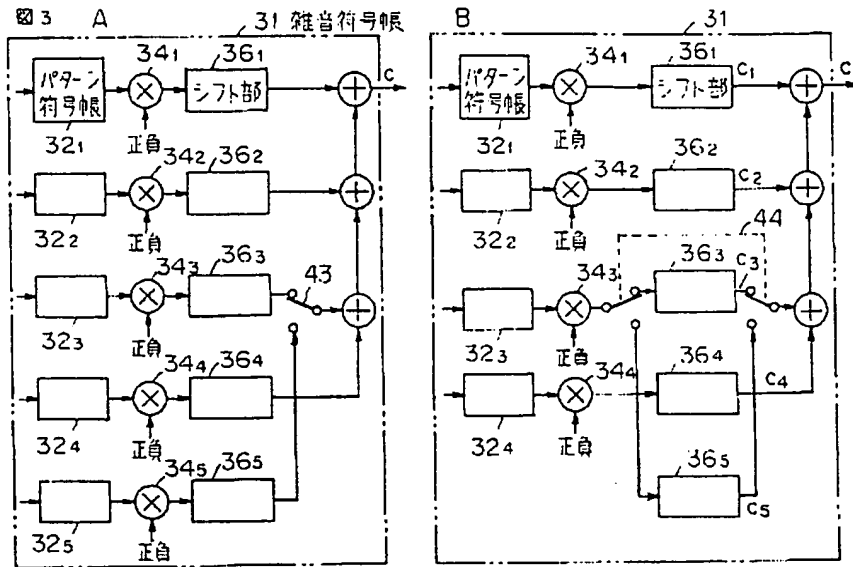
Figure 1 is a block diagram of a linear prediction parameter estimation device. The diagram is divided into several main sections:

- Input Section:**
 - 1: 原音 (Original Sound)
 - 2: 雑音符号帳 (Noise Symbol Table)
 - 3: 線形予測パラメータ符号化部 (Linear Prediction Parameter Symbol Encoding Section)
- Processing Section (Left):**
 - 31: パターン符号帳 (Pattern Symbol Table)
 - 32: サイン符号 (Sine Symbol)
 - 33: 位置符号 (Position Symbol)
 - 34: 乗算 (Multiplication)
 - 35: シフト部 (Shift Section)
 - 36: 加算 (Addition)
 - 37: 乗算 (Multiplication)
 - 38: 加算 (Addition)
 - 39: 乗算 (Multiplication)
 - 40: 加算 (Addition)
 - 41: 乗算 (Multiplication)
 - 42: 加算 (Addition)
 - 43: 乗算 (Multiplication)
 - 44: 加算 (Addition)
 - 45: 乗算 (Multiplication)
 - 46: 加算 (Addition)
 - 47: 乗算 (Multiplication)
 - 48: 加算 (Addition)
 - 49: 乗算 (Multiplication)
 - 50: 加算 (Addition)
 - 51: 乗算 (Multiplication)
 - 52: 加算 (Addition)
 - 53: 乗算 (Multiplication)
 - 54: 加算 (Addition)
 - 55: 乗算 (Multiplication)
 - 56: 加算 (Addition)
 - 57: 乗算 (Multiplication)
 - 58: 加算 (Addition)
 - 59: 乗算 (Multiplication)
 - 60: 加算 (Addition)
 - 61: 乗算 (Multiplication)
 - 62: 加算 (Addition)
 - 63: 乗算 (Multiplication)
 - 64: 加算 (Addition)
 - 65: 乗算 (Multiplication)
 - 66: 加算 (Addition)
 - 67: 乗算 (Multiplication)
 - 68: 加算 (Addition)
 - 69: 乗算 (Multiplication)
 - 70: 加算 (Addition)
 - 71: 乗算 (Multiplication)
 - 72: 加算 (Addition)
 - 73: 乗算 (Multiplication)
 - 74: 加算 (Addition)
 - 75: 乗算 (Multiplication)
 - 76: 加算 (Addition)
 - 77: 乗算 (Multiplication)
 - 78: 加算 (Addition)
 - 79: 乗算 (Multiplication)
 - 80: 加算 (Addition)
 - 81: 乗算 (Multiplication)
 - 82: 加算 (Addition)
 - 83: 乗算 (Multiplication)
 - 84: 加算 (Addition)
 - 85: 乗算 (Multiplication)
 - 86: 加算 (Addition)
 - 87: 乗算 (Multiplication)
 - 88: 加算 (Addition)
 - 89: 乗算 (Multiplication)
 - 90: 加算 (Addition)
 - 91: 乗算 (Multiplication)
 - 92: 加算 (Addition)
 - 93: 乗算 (Multiplication)
 - 94: 加算 (Addition)
 - 95: 乗算 (Multiplication)
 - 96: 加算 (Addition)
 - 97: 乗算 (Multiplication)
 - 98: 加算 (Addition)
 - 99: 乗算 (Multiplication)
 - 100: 加算 (Addition)
- Processing Section (Right):**
 - 7: 適応符号帳 (Adaptive Symbol Table)
 - 8: 同期符号 (Synchronization Symbol)
 - 9: 乗算 (Multiplication)
 - 10: 乗算 (Multiplication)
 - 11: 重み符号帳 (Weight Symbol Table)
 - 12: 加算 (Addition)
 - 13: 線形予測パラメータ復号化部 (Linear Prediction Parameter Decoding Section)
 - 14: 合成フィルタ (Synthesis Filter)
 - 15: 歪み計算部 (Distortion Calculation Section)
 - 16: 符号帳検索制御部 (Symbol Table Search Control Section)
 - 17: 符号送出部 (Symbol Output Section)
- Legend:**
 - 雑音符号 (Noise Symbol)
 - パターン符号 (Pattern Symbol)
 - 位置符号 (Position Symbol)
 - サイン符号 (Sine Symbol)
 - 同期符号 (Synchronization Symbol)
 - 雑音符号 (Noise Symbol)
 - 重み符号 (Weight Symbol)
 - 線形予測パラメータ符号 (Linear Prediction Parameter Symbol)
 - 歪み符号 (Distortion Symbol)
 - 符号送出部 (Symbol Output Section)

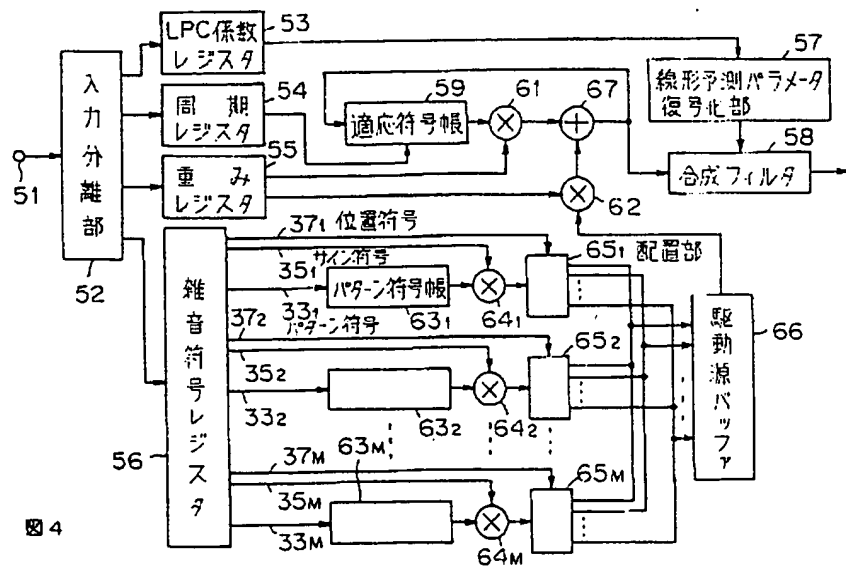
图 2



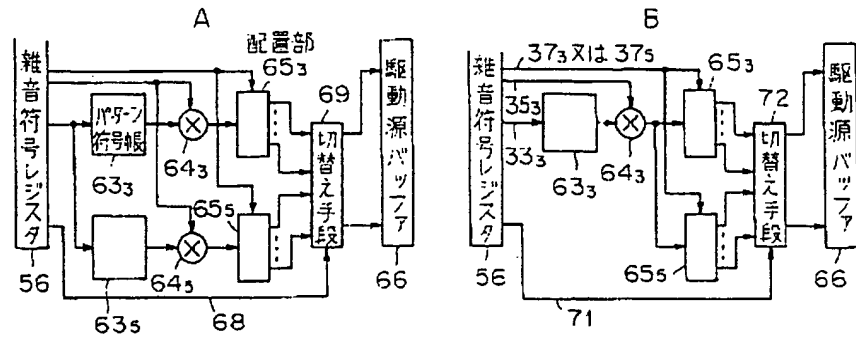
【圖3】



【図4】



【図5】



C

	ASELP方式	本発明による方法			
	17ビット	21ビット	18ビット	17ビット	
演算時間	4.4秒	2.0秒	1.5秒	1.2秒	
SN比	11.82dB	12.08dB	11.71dB	11.52dB	

図5

【図6】

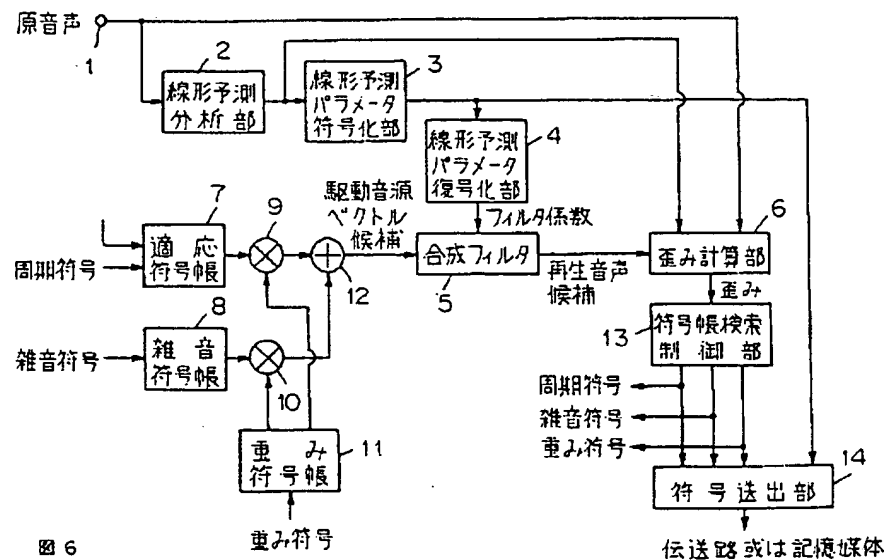


図6

[illegible]

図 8

A

原音声 1

聴覚重み付けフィルタ 16

合成フィルタ 21

聴覚重み付けフィルタ 22

聴覚重み付けフィルタ 24

合成フィルタ 23

適応符号帳 7

雑音符号帳 8

歪み評価尺度計算部 19

符号帳検索制御部 13

加算器 28

乗算器 9

乗算器 10

乗算器 26

乗算器 25

B

インパルス応答算出部 26

たたみ込み部 25

図 8 は、音声処理装置の構成図を示す。図 A は、音声処理の主要なブロックを示す。原音声 1 は、聴覚重み付けフィルタ 16 に入力され、出力 x_p は加算器 28 に加えられる。また、適応符号帳 7 と雑音符号帳 8 は、乗算器 9 と乗算器 10 にそれぞれ入力され、乗算器 9 の出力 g_a は合成フィルタ 21 に、乗算器 10 の出力 c は合成フィルタ 23 にそれぞれ入力される。合成フィルタ 21 と 23 の出力は、聴覚重み付けフィルタ 22 と 24 にそれぞれ入力される。聴覚重み付けフィルタ 22 の出力 a は加算器 28 に加えられる。聴覚重み付けフィルタ 24 の出力 a_q は、乗算器 26 に加えられる。乗算器 26 の出力 $g_r c$ は、たたみ込み部 25 に入力される。たたみ込み部 25 の出力 $g_r H C$ は、乗算器 25 に加えられる。乗算器 25 の出力は、歪み評価尺度計算部 19 に加えられる。歪み評価尺度計算部 19 の出力は、符号帳検索制御部 13 に送られる。図 B は、乗算器 26 とたたみ込み部 25 の詳細な構成を示す。乗算器 26 は、インパルス応答算出部 26 に入力される。インパルス応答算出部 26 の出力は、たたみ込み部 25 に送られる。たたみ込み部 25 は、たたみ込み演算を実行する。

(72)発明者 片岡 章俊
東京都千代田区内幸町1丁目1番6号 日本電信電話株式会社内